CHAPTER

8

NETWORKING BASICS

After reading this chapter and completing the exercises, you will be able to:

- List and describe the seven layers of the OSI model
- Understand the role of protocols and services in networks
- Explain the differences between a normal operating system and a network operating system
- List the differences between client and server roles

The advent of networks is one of the best (or debatably, the worst) things to happen to computers. Networks open up options that did not exist a relatively short time ago. This chapter looks at some of the networking basics, as well as the Open Systems Interconnection (OSI) model for networking. The OSI model is a set of guidelines for making all operating systems and system components work together in an open infrastructure. The chapter also covers the role of protocols and services in Windows 2000 Server.

Principles of Networking

What is a **network**? In its most rudimentary form, a network consists of two or more computers connected so that data can be transferred between them. Most of the networks that you encounter will, however, be much more complex than this model.

In the past, the first networks to which most businesses were exposed were known as "sneaker-nets." Transferring data from one computer to another in a sneaker-net network was a simple matter of placing the required data on a floppy disk, walking over to the receiving computer, and copying the data from the floppy disk to the second computer's hard drive. Setting up these networks was easy, expanding them was easier, and failures were almost unknown. Since the time of sneaker-nets, networks have grown by leaps and bounds. The following sections cover the basic types of modern networks.

LANs, WANs, and MANs

There are basically three types of networks: **local area networks (lans*)***, **wide area networks (wans)**, and **metropolitan area networks (mans)**.

A LAN is a network that is confined to a single location, such as an office, a floor in a building, or an entire office building. LANs usually use high-speed connections between the different systems—for example, 10 or 100 Mbps Ethernet, Asynchronous Transfer Mode (ATM), or Fiber Distributed Data Interface (FDDI). The actual physical connection is usually owned by the organization that uses the network.

WANs are networks that are connected over great distances, such as across a state, throughout a country, or between countries and continents. WANs are connected using slower connections that usually rely on telecommunications companies for the physical medium.

The third network type, a MAN, is similar to a WAN except that it connects locations within a city rather than between cities. This term is rarely used today, as most organizations use the term "WAN" to describe both intracity and intercity networks.

OSI Reference Model

As businesses around the world began to embrace the idea of networks, it became apparent that some sort of standard or guideline was necessary to allow separate and heterogeneous networks to communicate efficiently. In 1977, the International Organization for Standardization (ISO), a committee representing a wide range of organizations, developed a new standard for networking. This standard is the Open Systems Interconnection (OSI) reference model. The OSI model is nothing more than a set of guidelines for companies to use when developing network products (including network cards, routers, hubs, and **protocols**).

The OSI model consists of seven distinct levels, also called layers; each layer describes how its part in the communication scheme should function. In addition, each layer is also assigned a number from 1 to 7. Table 8-1 lists the seven layers and their associated numbers.

Table 8-1 OSI model layers

Layer name	Layer number
Application layer	Layer 7
Presentation layer	Layer 6
Session layer	Layer 5
Transport layer	Layer 4
Network layer	Layer 3
Data Link layer	Layer 2
Physical Layer	Layer 1

The OSI model allows software manufacturers to develop products that work and communicate at one or more layers within the model, with the assurance that those products will function with products developed by other vendors that conform to the model. For example, a network card manufacturer that develops an Ethernet card knows that the TCP/IP protocol written to conform to the OSI model in Windows 2000 will function after the appropriate driver is installed on the system.

In the following sections, we look at each of the OSI layers and examine its role and functions in the OSI model.

Tip

An easy way to remember the name of the layers in the OSI model is to use the anagram: **P**lease **D**o **N**ot **T**hrow **S**ausage **P**izza **A**way.

Application Layer

Most people initially assume that the **Application layer** refers to a user application, such as Microsoft PowerPoint or Word. In fact, this interpretation is not the case. The Application layer is responsible for the actual communication between a program that runs on the system and the network resources that the program accesses (for example, loading a Microsoft PowerPoint file from a shared location on a remote server).

Presentation Layer

The **Presentation layer** is responsible for determining the format of the data that are sent on the network. The remote system might exist in a different country and use a different character set, so a standard format that all systems understand is needed. The sending computer translates the format of its Application layer into this standard format, and the receiving computer translates the standard format into a format that is understood by its Application layer.

The most commonly used component that resides at this level is a **redirector**. A redirector simply captures the input or output of an application and redirects it to a different location.

This ability allows you to save a file to the G: drive when it is actually a mapped drive to a share on a remote server's hard drive, for example. The Presentation layer is also responsible for data compression, encryption, and decryption, as well as protocol and character set conversion.

Session Layer

The **Session layer** is responsible for establishing and maintaining connections between communicating systems. This layer ensures that all data are transferred to the correct application and the correct system when multiple communication streams are active. You can compare the Session layer to making a phone call. The telecommunications company that supplies you with your phone service has a switch that connects you to the correct number when you dial it. The company's equipment must then maintain that connection until you or the person at the other end of the line disconnects. Telephone companies manage thousands of simultaneous connections, and, depending on the size of your network, the Session layer may have to do the same.

Transport Layer

Under most circumstances, the data you send over the network are larger than a single **frame**, also called a data packet. The **Transport layer** is responsible for breaking the data up into smaller sections so that the pieces can be sent in individual frames. To ensure that the receiving computer's Transport layer can reassemble the data, the sending computer's Transport layer assigns a sequence number to each frame. The receiving computer's Transport layer makes sure that the frames are reassembled into the data stream in the correct order. A **cyclic redundancy check (CRC)** is also performed on the frames to ensure that they have been received error-free.

Network Layer

The **Network layer** controls the way that data are routed from one system to another. A routable protocol needs to have an addressing scheme that specifies not only its local address, but also its global address. You can compare this addressing scheme to telephone area codes. Several locations (networks) may have the same phone number (local address), but what distinguishes the same number (local address) in city A (network A) from that in city B (network B) is the area code (global address) that precedes the number. This layer also figures out the best path that the data should traverse to get from the sending system to the receiving system.

Data Link Layer

The **Data Link layer** is responsible for connecting the physical media of the layer below it (the copper and electrical pulses from the Physical layer) and the data of the layers above it.

It is at this layer that **hardware addresses** are used. Hardware addresses are also known as **Media Access Control (MAC) addresses**.

> Tip
>
> MAC addresses consist of six hexadecimal bytes (ranging from 00 to FF). These bytes are usually written with either a dash (-) or a colon (:) separating them—or example, 12:34:56:78:90:AB (or AB-CD-EF-12-34-56). The first three octets represent a unique number that is assigned to each company that produces network cards and devices and are known as the official Institute of Electrical and Electronic Engineers (IEEE) Organizationally Unique Identifier (OUI). Each organization must assign a unique value to each device that it produces. That value must use the last three bytes. For more information on the currently assigned company prefixes and the process of applying for one, visit the IEEE's OUI Web page at *http://standards.ieee.org/regauth/oui/index.html*.

Physical Layer

The **Physical layer** defines an actual physical network device, which can include the type of connectors, the type of media (copper and fiber), and electrical voltage. In effect, the Physical layer takes the signal, amplifies it, and sends it down the "wire." The components at the Physical layer do not look at the data within each of the frames; they do not care what the data include or where the information might be heading.

Putting It All Together

Each layer in the OSI model describes how its part in the communication scheme should function. It is important to understand that communication can take place only between layers that are in direct contact with each other (that is, directly above or below each other). For example, the Application layer can talk only to the Presentation layer, the Presentation layer can talk to either the Application or the Session layer, the Session layer can talk to either the Presentation or the Transport layer, and so on. Layers on different systems can also communicate—for example, the Session layer on system A can communicate with the Session layer on system B. This communication must "flow" through the OSI model on both systems—except in the Physical layer; only the Physical layers on separate networks can communicate with each other directly (see Figure 8-1).

To envision the data flow between the OSI model layers, imagine that there are two adjacent seven-story buildings (building A and building B) both owned by the company where you work as a mail clerk. The departments on each floor of each building correspond to one another. That is, the names of the departments (listed in Table 8-2) on each floor of both buildings are the same.

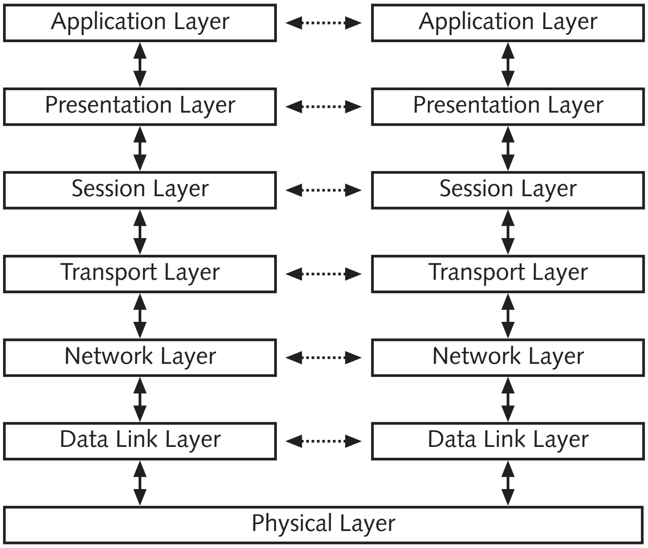


Figure 8-1 The OSI model

Table 8-2 Department names for buildings A and B

Floor	Department
7	Accounting
6	Purchasing
5	Sales
4	Telecommunication
3	Personnel
2	Lobby
1	Parking Garage

Every morning you start your rounds in building A on the 7th floor in the Accounting department. You pick up all Accounting department mail and then descend to the Purchasing department on the 6th floor, where you pick up all of that department's mail. This pattern continues until you finally reach the parking garage (which connects the two buildings). You walk over to building B and repeat the process in reverse, starting in the lobby and working your way up to the Accounting department on the 7th floor. Company policy states that mail can be transferred only from one department to its sister department in the other building (Accounting A to Accounting B, Purchasing A to Purchasing B, and so on).

The OSI model works in much the same way. The information you send (that is, the data) is analogous to the mail cart that you use to carry the mail. As the data pass by each layer (department), that layer (department) attaches a component to the data (puts its mail in the cart) and passes it down. At the receiving end, each layer (department) strips the component (mail) placed on the data (cart) by its sister layer and passes it up the layers until the Application layer passes only the data to the user programs running on the computer.

ROLE OF PROTOCOLS AND SERVICES

Before computers can communicate with one another over a network, they need to "speak" the same language. This common language is known as a protocol. The protocol is charged with answering following questions:

- Will communications use data compression? If so, what type of compression?
- How do computers on different networks communicate?
- How does the receiving computer know that all of the data has been sent?
- How does the receiving computer notify the sending computer that it has received all of the data?
- Will communications use error correction?
- How many frames should a computer receive before acknowledging them?

The protocol that you use on your network really depends on the network's infrastructure. Protocols can be classified into two types: local and remote. A local protocol is one that can be used only in a single network environment and is known as a **nonroutable protocol**. A remote protocol, also known as a **routable protocol**, functions in local environments and can communicate with remote networks. Nonroutable protocols tend to be faster. In a single network environment, there is no need to worry about duplicate addresses, maintenance of a list of routes from one network to another, and name-to-address translations.

Currently, three main protocols are used on networks: **NetBIOS Enhanced User Interface (NetBEUI)**, **Internetwork Packet Exchange/Sequenced Packet Exchange (IPX/SPX)**, and **Transmission Control Protocol/Internet Protocol (TCP/IP)**. NetBEUI is a nonroutable protocol, and IPX/SPX and TCP/IP are routable protocols. Microsoft designed Windows 2000 so that it depends on one of these protocols—TCP/IP. Each of these protocols is discussed in the following sections.

NetBEUI

NetBEUI is actually an acronym within an acronym: It stands for NetBIOS Enhanced (or Extended) User Interface, and NetBIOS in turn stands for Network Basic Input/Output System. IBM developed this protocol in the early 1980s, and Novell and Microsoft later adopted it. Currently, it is most commonly found in small, single-network, Microsoft-based environments. NetBEUI cannot be used in a WAN environment because it is nonroutable; for the same reason, however, it is also extremely easy to set up and very fast. In fact, it is the fastest of the three protocols.

Do not confuse NetBEUI and NetBIOS. NetBEUI is a transport protocol, whereas NetBIOS is an API.

IPX/SPX

IPX/SPX is Novell's networking protocol. Until the release of Novell's NetWare 5, all NetWare environments required IPX/SPX. (NetWare 5 also supports TCP/IP.) Microsoft's operating systems use a version of this protocol called NWLink, which is an IPX/SPX-compatible protocol. IPX/SPX is actually a suite of protocols. In this suite, SPX is a Transport layer protocol that runs on top of IPX (a Network layer protocol).

IPX/SPX uses the MAC address of each system on the network as its network address. Another address is also given to the entire network. This scheme allows IPX/SPX to be routed. IPX/SPX is used only on internal networks, however, because most Internet routers do not support it.

TCP/IP

The Internet uses the TCP/IP protocol exclusively, leading to the last part of the expansion of the acronym—Internet Protocol. TCP/IP is often referred to as the TCP/IP protocol suite, because most systems support both the TCP/IP protocol and the applications that go with it. Without this particular protocol, the Internet would not exist in its present, familiar form.

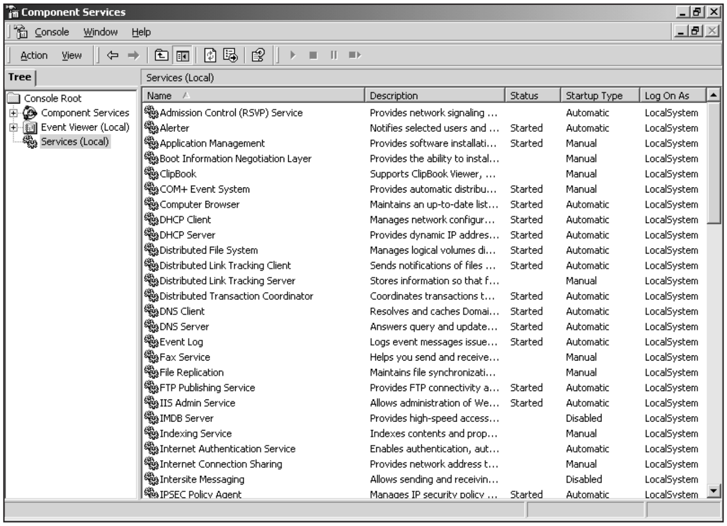
Understanding TCP/IP is a must when dealing with Windows 2000 systems, because most Windows 2000 features rely on it to work. TCP/IP domains have replaced Windows NT domains, which allows the close integration of Windows 2000 networks with the Internet and permits the networks to interact fully with all Internet technologies.

Although TCP/IP is the most widely used protocol today, it is also the slowest of the three protocols mentioned in this section. TCP/IP was originally designed as a protocol that would allow a network developed by the U.S. military to recover from catastrophic failures. That network eventually evolved into the Internet. The original idea behind the Internet was the creation of a network that would enable military installations to communicate even if one or more of these sites became destroyed in a war. Because the TCP/IP protocol was designed to ensure that devices (now known as routers) could decide on the best path to send the data so that the information was received, it needed flexible addressing schemes and error correction. These addressing schemes and error correction add overhead to devices using the protocol, making TCP/IP very slow, but also highly robust.

Services

Services are server-based applications that run on a server. They tend to run in the background and give the operating system its functionality. Windows 2000 includes services for logon and authentication, computer naming, addressing, and Web and File Transfer Protocol (FTP), as well as many others. In previous versions of Windows NT, services were accessed through the Control Panel; in Windows 2000, you access the services using the Component Services application. To access these services, follow these steps:

1. Click on Start, Programs, Administrative Tools, Component Services.
2. Select the Services (local) option in the left pane. The services installed on the system will appear in the right pane, as shown in Figure 8-2.



Services (Local)

Name	Description	Status	Startup Type	Log On As
Admission Control (RSVP) Service	Provides network signaling ...		Automatic	LocalSystem
Alerter	Notifies selected users and ...	Started	Automatic	LocalSystem
Application Management	Provides software installati...	Started	Manual	LocalSystem
Boot Information Negotiation Layer	Provides the ability to instal...		Manual	LocalSystem
ClipBook	Supports ClipBook Viewer, ...		Manual	LocalSystem
COM+ Event System	Provides automatic distribu...	Started	Manual	LocalSystem
Computer Browser	Maintains an up-to-date list...	Started	Automatic	LocalSystem
DHCP Client	Manages network configur...	Started	Automatic	LocalSystem
DHCP Server	Provides dynamic IP addres...	Started	Automatic	LocalSystem
Distributed File System	Manages logical volumes di...	Started	Automatic	LocalSystem
Distributed Link Tracking Client	Sends notifications of files ...	Started	Automatic	LocalSystem
Distributed Link Tracking Server	Stores information so that f...		Manual	LocalSystem
Distributed Transaction Coordinator	Coordinates transactions t...	Started	Automatic	LocalSystem
DNS Client	Resolves and caches Domai...	Started	Automatic	LocalSystem
DNS Server	Answers query and update...	Started	Automatic	LocalSystem
Event Log	Logs event messages issue...	Started	Automatic	LocalSystem
Fax Service	Helps you send and receive...		Manual	LocalSystem
File Replication	Maintains file synchronizati...		Manual	LocalSystem
FTP Publishing Service	Provides FTP connectivity a...	Started	Automatic	LocalSystem
IIS Admin Service	Allows administration of We...	Started	Automatic	LocalSystem
IMDB Server	Provides high-speed access...		Disabled	LocalSystem
Indexing Service	Indexes contents and prop...		Manual	LocalSystem
Internet Authentication Service	Enables authentication, aut...		Automatic	LocalSystem
Internet Connection Sharing	Provides network address t...		Manual	LocalSystem
Intersite Messaging	Allows sending and receivin...		Disabled	LocalSystem
IPSEC Policy Agent	Manages IP security policy ...	Started	Automatic	LocalSystem

Figure 8-2 The Component Services administrative tool

To configure a service's properties, you double-click the service name in the right pane of the Component Services application. You can also access the service's properties by choosing the Properties option from the Action menu. A service's properties are classified into four categories:

- General settings
- Logon preferences
- Recovery options
- Service dependencies

In the following sections, we discuss each of these service configuration properties.

General Settings

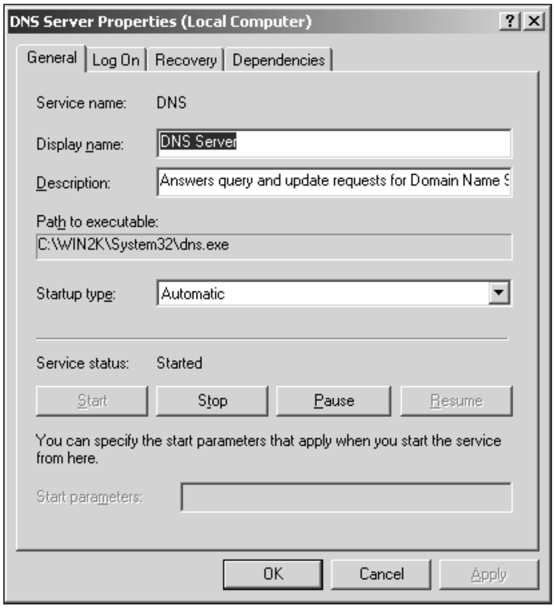
The General tab of the service properties allows you to modify the default display name (the name by which the service is referred to in the Component Services application) and the service's description. (Figure 8-3 shows the General tab of the DNS Server service properties.) It allows you to customize the look and feel of your service administration tool. Notice that the path to the service is listed on this tab as well, which helps you to troubleshoot your services. If a service executable becomes corrupted, you can find the executable file for the service, note its creation date and size, and copy the file from another installation of Windows 2000 that uses the same file information.

The next two sections in the General properties tab are Startup type and Service status. The startup type defines how (or if) Windows starts the service during its initialization procedure.

- *Startup type:* Three service startup types exist: automatic, manual, and disabled. When a service is given an automatic startup type, it starts when Windows 2000

Server initializes and starts the services. Under the manual startup type, the service does not start during the Windows 2000 initialization procedure, but can be started by either a dependent service or a user. With the disabled startup type, the service cannot be started by Windows 2000 initialization, a dependent service, or a user. This last option is useful when a service is temporarily not required and disabling it will free up system resources that would normally be assigned to this particular service.

- *Service status:* This item indicates the current status of a service. The service can be in one of three states: started, stopped, or paused. A started service has been executed and is currently running on the system. A stopped service is not running and does not respond to any requests. A paused service does not accept any new connections unless the user attempting the connection is a member of either the Administrator or Server Operators group. You must click the Resume button if a service is paused to enable users to connect to the service again.



Service name: DNS

Display name: DNS Server

Description: Answers query and update requests for Domain Name S

Path to executable:
C:\WIN2K\System32\dns.exe

Startup type: Automatic

Service status: Started

You can specify the start parameters that apply when you start the service from here.

Start parameters:

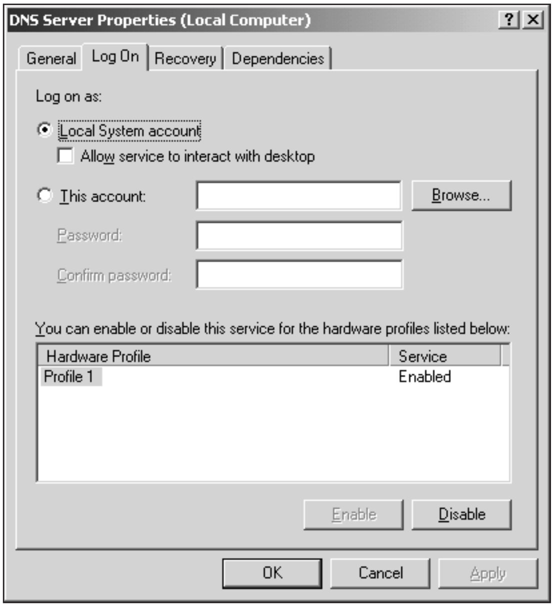
Figure 8-3 Services properties window

Log On Preferences

Most services use the local system account to access the operating system. That is, when the service needs to request or send information to the operating system, it uses the authentication properties of an internal account. This account cannot be modified and is recognized by the operating system as having the appropriate permissions to perform most system tasks. In some instances, you might want to create a dedicated account for the service to use in performing its tasks. Usually, the applications that require the service must be aware of the account name and the password before they can run successfully. An example of such an application is Microsoft Exchange. Exchange uses an account called the Exchange Service Account to log into the system and obtain information about users. If you change the user name or password for this service without notifying the Exchange Administrator program, Exchange will fail when it tries to access the system.

Windows 2000 also offers the ability to control which services are started for a specific profile. In the past, with Windows NT, users could control which devices and device drivers were started and used with a specific profile, but they lacked an easy way to control individual services. The ability to control which services are started for a specific profile is useful when testing a system or dealing with a system that is used as a backup. To understand how this ability works, consider the following scenario: Your organization is running a Windows 2000 Server as a domain controller. You need to bring it down occasionally for servicing, but your company cannot afford to purchase a second system to act as a backup. You can install Windows 2000 Server on a desktop system and create a secondary profile that starts the server as a domain controller. When you need to service the server, you simply reboot the system and choose the secondary profile. Although the system might not run as efficiently as the server would, it will run.

Figure 8-4 shows the Log On tab.



Log on as:

● Local System account

□ Allow service to interact with desktop

○ This account:

Password:

Confirm password:

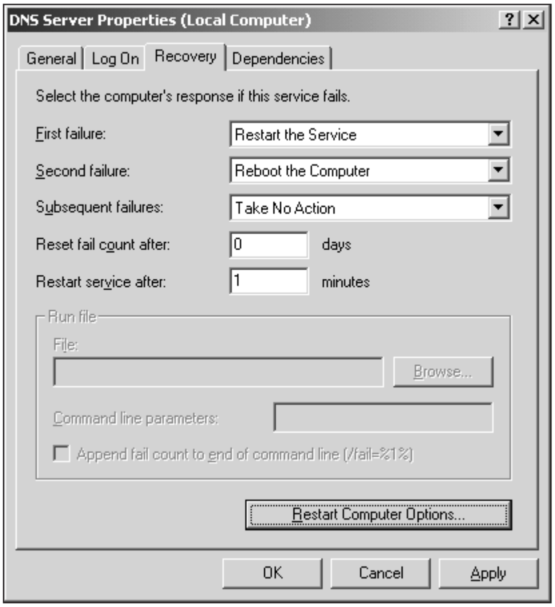
You can enable or disable this service for the hardware profiles listed below:

Hardware Profile	Service
Profile 1	Enabled

Figure 8-4 Service Properties, Log On tab

Recovery Options

Another Windows 2000 Server feature is the ability to dynamically respond to service failures. By default, if a service fails, Windows 2000 Server will not react. You can, however, configure services so that they will perform tasks that you specify in the event of a failure. This type of definition can be made for all services or for only selected ones. As shown in Figure 8-5, you have three levels of failure: first, second, and subsequent failures. When choosing which action Windows 2000 Server will take if a service fails, you can opt to take no action (the default option on all failures), to restart the service, to run a file, or to reboot the computer. All of these actions are fairly self-explanatory, except perhaps for the "run a file" option. With this option, you can have Windows 2000 Server execute an external file that sends an e-mail or pages an administrator, notifying him or her of the error.



DNS Server Properties (Local Computer)

Select the computer's response if this service fails.

First failure: Restart the Service

Second failure: Reboot the Computer

Subsequent failures: Take No Action

Reset fail count after: 0 days

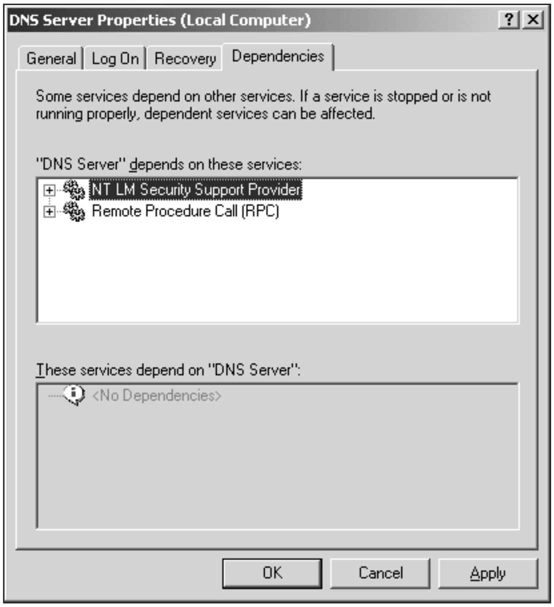
Restart service after: 1 minutes

□ Append fail count to end of command line (/fail=%1%)

Figure 8-5 Recovery options for a service

Service Dependencies

The last tab in the service properties is Dependencies (see Figure 8-6). This tab is strictly informational and cannot be modified. The services on which the selected service depends are listed in the top window (if any), whereas the services that depend on the selected service appear in the bottom window (if any). When service A depends on service B, service B must be running before service A can start. If service B is set to either an automatic or manual startup type, either the operating system or service A will start it before starting itself.



Some services depend on other services. If a service is stopped or is not running properly, dependent services can be affected.

"DNS Server" depends on these services:

- NT LM Security Support Provider
- Remote Procedure Call (RPC)

These services depend on "DNS Server":

- <No Dependencies>

Figure 8-6 Service dependencies window

NORMAL AND NETWORK OPERATING SYSTEM BASICS

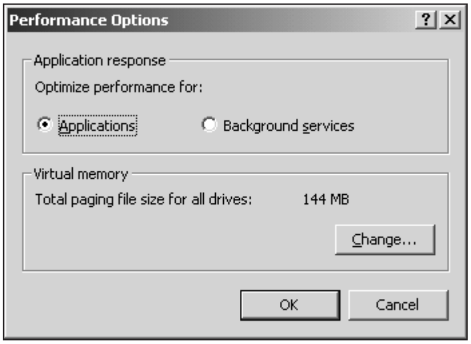
On the surface, it may appear as if many network operating systems are not much different from their desktop counterparts. In reality, nothing could be further from the truth. Although the Windows 2000 Server interface looks like the Windows 95/98 interface, the operating system is not the same. Both Windows NT and Windows 2000 Server are 32-bit operating systems that were written for networking from the ground up. As a result, they do not have some of the limitations of the Windows 95/98 operating systems. Most of today's operating systems have the ability to share printers and files as well as to run FTP or Web services. Consequently, it is important to understand the differences between a normal operating system and a network operating system (NOS).

A network operating system is designed to work with network services first and desktop applications second, whereas the opposite is true for a traditional (normal) operating system. Although most network operating systems can run applications such as word processor or spreadsheet programs, some (for example, Novell's NetWare) do not. Network operating systems are written to serve information to clients first and run applications second. In contrast, normal operating systems focus on running applications locally.

You can configure Windows 2000 Server to allocate a portion of system resources to background applications (such as services) and another portion to foreground applications. To access this configuration option, follow these steps:

1. Right-click My Computer and choose Properties from the drop-down menu.
2. Click the Advanced tab in the System Properties dialog box.

3. Click the Performance Options button. A window similar to the one shown in Figure 8-7 appears. At this point, you can choose the Applications radio button to optimize Windows 2000 Server to run foreground applications or the Background Services radio button to optimize it to run the services. The latter is the default configuration when you install Windows 2000 Server.
4 Click OK to close the Performance Options dialog box.
5. Close the System Properties dialog box.


Performance Options

Application response
Optimize performance for:
● Applications ○ Background services

Virtual memory
Total paging file size for all drives: 144 MB

Figure 8-7 Windows 2000 Server performance configuration

In general, network operating systems support more high-end hardware configurations, such as symmetric multiprocessing (the ability to use multiple processors in a single system), extremely large hard disk support, and fault-tolerant storage systems (such as SCSI and fiber-channel solutions). You will also find that a network operating system gives you better control over network security.

Windows 2000 Networking Model

The best way to describe the Windows 2000 networking model is to explain how it corresponds to the OSI model (starting from the Application layer at the top). At the Application layer, Windows has *providers*. These providers create a way for the applications running on the system to communicate with the network.

The line between the Application and Presentation layers in the OSI model corresponds to the one between the User mode and the Kernel mode in the Windows model. All executive services run in the Presentation layer.

Redirectors exist at the Session layer. These redirect any requests for information from a physical device to a logical one. For example, when an application saves a file to a network drive rather than a physical one, the redirector sends the information to the correct location.

At the Transport and Network layers, you will find the Transport protocols, or the languages that are used to communicate on the network. Connecting these protocols to the redirectors is the **Transport Driver Interface (TDI)**. This layer allows the protocols to remain independent of the redirectors.

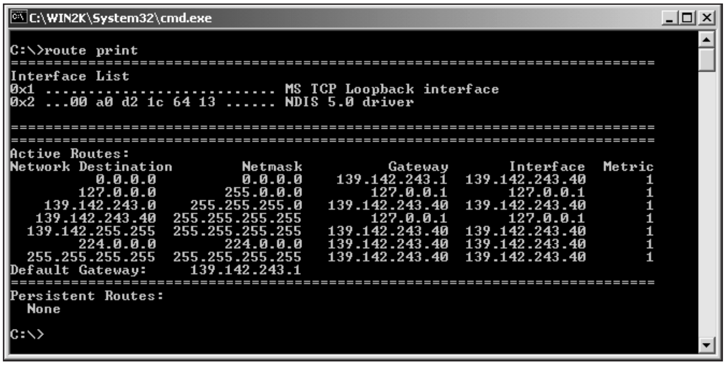
At the Data Link layer, the network interface card drivers exist. These drivers allow the system to communicate with the physical network cards. The NDIS interface appears between these drivers and the Transport protocols. Like the TDI, this interface allows the drivers to communicate with the transport protocols above without the need to rewrite any code.

ROUTING AND REMOTE ACCESS

A major strength of the TCP/IP protocol is its ability to address a huge number of addresses and networks. The process of transferring or forwarding packets of data from one network to another is known as **routing**. If your network spans a single location, then you probably will not encounter routing. If your network spans more than a single location or needs to communicate with a WAN or the Internet, however, then routing becomes an essential component of the network.

IP Routing

As stated earlier, routing is simply the transferring of packets from one network to another. When a computer attempts to route a packet between two networks, it checks its **routing table**. The routing table lists the available networks and the network interfaces over which the system must communicate to contact the remote network. Figure 8-8 depicts a routing table. Two types of routing exist: **static routing** and **dynamic routing**.



```
C:\>route print
===========================================================================
Interface List
0x1 ........................... MS TCP Loopback interface
0x2 ...00 a0 d2 1c 64 13 ...... NDIS 5.0 driver

===========================================================================
===========================================================================
Active Routes:
Network Destination        Netmask          Gateway       Interface  Metric
          0.0.0.0          0.0.0.0    139.142.243.1  139.142.243.40       1
        127.0.0.0        255.0.0.0        127.0.0.1       127.0.0.1       1
    139.142.243.0    255.255.255.0   139.142.243.40  139.142.243.40       1
   139.142.243.40  255.255.255.255        127.0.0.1       127.0.0.1       1
  139.142.255.255  255.255.255.255   139.142.243.40  139.142.243.40       1
        224.0.0.0        224.0.0.0   139.142.243.40  139.142.243.40       1
  255.255.255.255  255.255.255.255   139.142.243.40  139.142.243.40       1
Default Gateway:     139.142.243.1
===========================================================================
Persistent Routes:
  None

C:\>
```

Figure 8-8 A routing table

Static Routes

With a static route, the administrator is responsible for manually configuring all paths (or routes) from one network to another. If the route changes, the router will not notify other routers of the change—it is up to an administrator to manually modify the route. This method of configuring routes is good only for small networks or for devices that do not have the ability to broadcast their location (such as some print servers) and whose addresses do not change regularly.

Windows 2000 offers several ways to create and modify static routes. One new option in Windows 2000 for adding static routes is the Static Route dialog box. You can execute this dialog box from the Routing and Remote Access console (as seen in Figure 8-9).

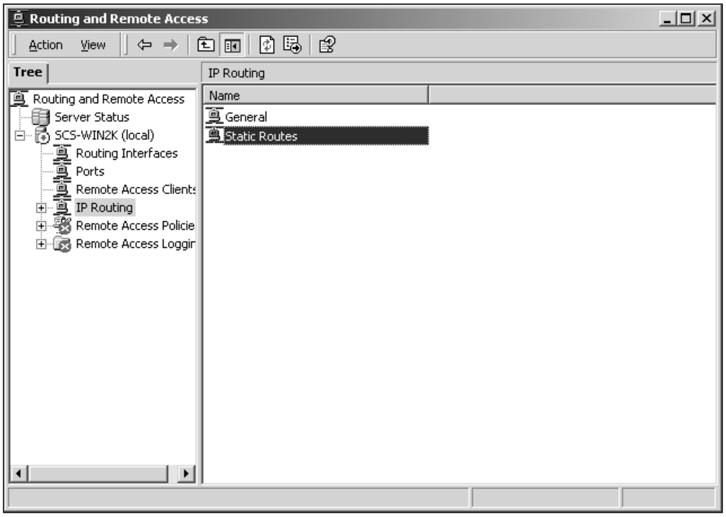
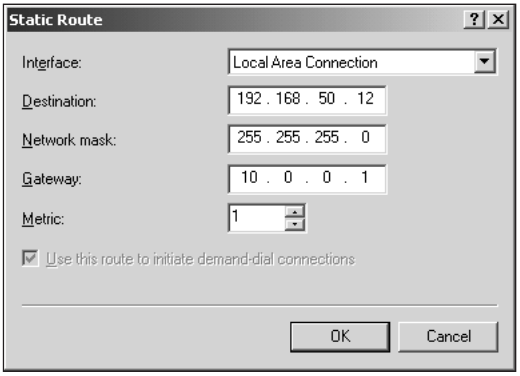


Figure 8-9 The Routing and Remote Access console

Two command prompt commands are available to you for configuring static routes: *route* and *ipkern*. The route command has been around for a while; you would issue a route add command to add a route. The ipkern command is new in Windows 2000; it is the command-prompt version of the Static Route dialog box (see Figure 8-10).


Static Route

Interface: Local Area Connection

Destination: 192 . 168 . 50 . 12

Network mask: 255 . 255 . 255 . 0

Gateway: 10 . 0 . 0 . 1

Metric: 1

☑ Use this route to initiate demand-dial connections

Figure 8-10 Static Route dialog box

Note

A static route can overwrite a dynamic route.

Dynamic Routes

With dynamic routing, all routing information is shared between the routers. When a router learns of a route to a network, it will pass that information along to the routers with which it communicates. The routers can therefore "learn" about the networks with which they communicate. This technique is the most common way of routing packets today. Windows 2000 supports two TCP/IP routing protocols: **Routing Information Protocol for Internet Protocol (RIP for IP)** and **Open Shortest Path First (OSPF)**.

Remote Access

Remote access allows you to connect remote clients to your network using a multitude of hardware devices. When a client connects to the remote access server, it becomes part of the network. It can browse files and folders, print to network printers, collaborate with other network users, and run applications. In Windows 2000, you use the Routing and Remote Access management console to configure remote access on a server (see Figure 8-11).

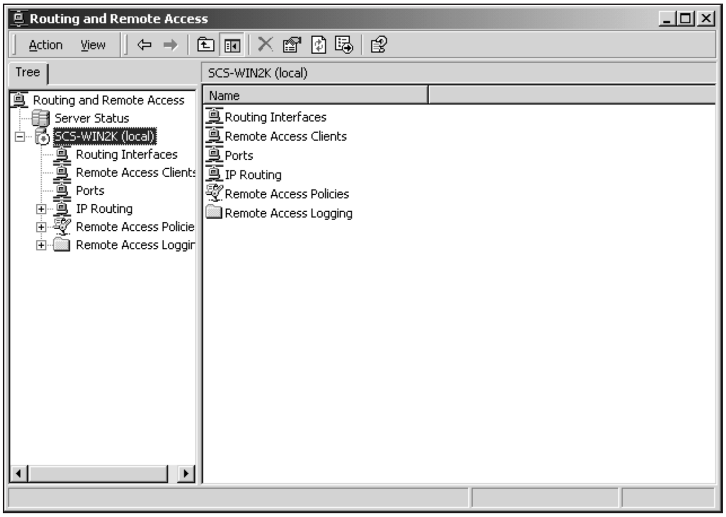


Figure 8-11 Routing and Remote Access console

Remote access includes two components, the client and the server. The client connects to the network, whereas the server acts as the gateway between the network and the remote client. To permit this type of communication, they need to share some protocols. Windows 2000 supports the following protocols for remote access:

- TCP/IP
- NetBEUI
- **AppleTalk Remote Access Protocol (ARAP)**
- NWLink IPX/SPX-compatible protocol

Windows 2000 also supports **virtual private networks (VPNs)**.VPNs add a second secure protocol to allow the client to use the Internet to connect to the network.With the regular remote access protocols, the client connects directly to the Windows 2000 Server. With a VPN, however, the client first connects to the Internet using its preferred Internet service provider.The client would then create a second session (also known as a **tunnel**) to a remote access server on the LAN, thereby producing an encrypted channel between the client and the server.The client can then participate on the network as if it were physically connected to the LAN.

VPNs can use either of two protocols to create these secure connections: **Point-to-Point Tunneling Protocol (PPTP)** and **Layer Two Tunneling Protocol (L2TP)**. PPTP is a technology that existed in Windows NT 4.0; L2TP is a protocol that is new to Windows 2000.

PPTP will encrypt all data that are sent between the client and the server. Secondary TCP/IP addresses are assigned to both the client and the server, and all data are routed using these "private" addresses. Anyone who captures the data stream between the two systems will have to decrypt the information to make it readable and useful.

L2TP resembles PPTP, except that it provides only the tunneling capability, not the encryption. All encryption is handled by a second method, such as IP Security (IPSec).

SERVER VERSUS CLIENT ROLES

It is important that you understand some of the different roles that a Windows 2000 server can play. A server typically can switch between client and server roles. As a rule, a *server role* can be defined as any task that is performed by the system for the benefit of another system (the client). A *client role* is defined as any task that the system requests from another, remote system.

Servers can play several client roles; from logging into the network, to printing to a network printer, to accessing files and applications on servers.The service that supports this flexibility is the Workstation Service.When this service is active, the server acts as a workstation. If you disable the Workstation Service, the system would no longer be able to connect to network resources.

In much the same way, servers use the Server Service to share files, printers, and other resources to clients. By disabling the Server Service, you effectively deny any client access to the network. This feature is a good way to stop clients from connecting to the network, if you need to make a modification to the server. Another common practice is to pause this service, rather than to stop it. Pausing the service maintains all existing connections to the server, while denying any new connections from being forged.

All Windows 2000 systems include both the Workstation and Server services. The Workstation Service on one system communicates with the Server Service on the other, and vice versa.

CHAPTER SUMMARY

- A network can be defined as a collection of computers that use a common language (protocol) to communicate. Networks allow you to share data, printers, and applications between users.
- The OSI model was developed to enable developers to create modular components that will "fit" together without having to rewrite all of the components for each device. Without such a standard, the entire networking suite would have to be replaced for any new application, network card, or protocol that was introduced.
- The seven distinct layers of the OSI model (Application, Presentation, Session, Transport, Network, Data Link, and Physical) and network protocols are basic to an understanding of computer networking. The three main protocols employed are NetBEUI, IPX/SPX, and TCP/IP. Protocols are the common languages that systems use to communicate. If one system uses a protocol that is different from that of another system, the two systems will not be able to communicate, even though they may be physically on the same network.
- Services are applications that run on servers and that allow components to run when no users are logged into the server. If services did not exist, applications would have to be executed while a user is logged in. These applications would be terminated if the user is logged out of the system.
- Network operating systems differ from desktop operating systems in that they are tuned to run on the "back end." As a result, they are better suited to running applications and services for other systems than running applications locally. For example, Microsoft Word would run considerably better on a Windows 2000 Professional system than on a Windows 2000 Server system, because the Professional system is designed to run Word, whereas the Server system is not.
- Routing allows Windows 2000 to connect two or more networks. Using some of the built-in services in Windows 2000, you can effectively convert a server into a router. The server will then transfer information between the networks.
- Remote access allows clients that are not connected directly to the LAN to connect to a Windows 2000 Server and participate on the network as if they were physically connected to it. This feature gives remote users access to network resources, such as printers, applications, and files.
- Servers tend to run both server and client components. Server components are geared toward servicing the requests of clients, whereas workstation components request information from a server.

Key Terms

AppleTalk Remote Access Protocol (ARAP) — A protocol that allows Apple Macintosh computers to connect to a remote access server.

Application layer — The layer of the OSI model that allows access to networking services.

cyclic redundancy check (CRC) — A mathematical recipe that generates a specific value, called a checksum, based on the contents of a data frame. The CRC is calculated before a data frame is transmitted and then is included with the frame; on receipt, the CRC is recalculated and compared with the sent value. If the two agree, the data frame is assumed to have been delivered intact; if they disagree, the data frame must be retransmitted.

Data Link layer — The layer of the OSI model that uses the hardware address of the system to communicate.

dynamic routing — The process used by routers to dynamically learn about the routes that they can take to connect to remote networks.

frame — The basic package of bits that represents a protocol data unit (PDU) sent from one computer to another across a network. In addition to its contents, a frame includes the sender's and receiver's network addresses as well as control information at the head and a CRC at the tail.

hardware address — See *Media Access Control (MAC) address.*

Internetwork Packet Exchange/Sequenced Packet Exchange (IPX/SPX) — A protocol developed by Novell for its NetWare operating system. It may be used in routed environments.

Layer Two Tunneling Protocol (L2TP) — A protocol that relies on other encryption methods (such as IPSec) for communication. It creates the secure connection, but other methods of encryption must be used.

local area network (LAN) — A group of computers that are connected to form a network within a small area, such as a floor or a building.

Media Access Control (MAC) address — A unique number that is assigned to each network device. It ensures that no two devices exist with the same addressing information.

metropolitan area network (MAN) — A network of computers that exist within the same metropolitan area, such as a city.

NetBIOS Enhanced User Interface (NetBEUI) — A protocol that can be used in small, nonrouted environments.

network — Two or more computers connected so that they can transfer information.

Network layer — The layer of the OSI model that addresses the messages for delivery.

nonroutable protocol — A network protocol that cannot be used in a routed network environment.

Open Shortest Path First (OSPF) — A protocol used by routers to learn about different routes to remote networks.

Organizationally Unique Identifier (OUI) — A unique number that is assigned to each network device vendor to ensure that hardware addresses do not overlap.

Physical layer — The layer of the OSI model that defines the physical structure of the network (copper, fiber, and so on).

Point-to-Point Tunneling Protocol (PPTP) — A protocol that is used to encrypt data between a server and a client.

Presentation layer — The layer of the OSI model that translates data from a format understood by the application into a generic format that can be understood by other systems.

protocol — A common language that allows heterogeneous systems to communicate and share information on a network.

redirector — An Application layer software component that captures application output and redirects it to a different location.

routable protocol — A network protocol that can be used in a routed environment to communicate with remote networks.

routing — The process of transferring packets of information from one network to another network.

Routing Information Protocol for Internet Protocol (RIP for IP) — A protocol used by routers to learn about different routes to remote networks.

routing table — A list of available networks and interfaces over which a system must communicate to contact a remote system.

service — A software component that exists on servers that run in the background so as to perform normal server operations, such as file and print sharing, Web and FTP services, and DNS services.

Session layer — The layer of the OSI model that initiates and maintains the communication between different systems on the network.

static routing — A system in which the network administrator must manually configure all paths from one network to another.

Transmission Control Protocol/Internet Protocol (TCP/IP) — The protocol for the Internet. It allows for the connection of large networks in different geographical locations.

Transport Driver Interface (TDI) — The specification to which all transport protocols must be written so that they can be used by higher-layer services, such as programming interfaces, file systems, and interprocess communication mechanisms.

Transport layer — The layer of the OSI model that is responsible for ensuring error-free transmission and reception of data.

tunnel — A communication mechanism used by VPNs to establish a second, secure session between a client and remote server.

virtual private network (VPN) — A secure connection between a client and a private network over the Internet.

wide area network (WAN) — A group of computers that are networked over great distances, such as between cities.

Review Questions

1. The OSI model is a set of rules and standards that vendors must follow. True or False?
2. Only the Physical layer of the OSI model communicates directly with the Physical layer on another network. True or False?
3. At which OSI layer is error checking added?
   a. Physical layer
   b. Data Link layer
   c. Network layer
   d. Transport layer
   e. Session layer
   f. Presentation layer
   g. Application layer
4. At which OSI layer does compression take place?
   a. Physical layer
   b. Data Link layer
   c. Network layer
   d. Transport layer
   e. Session layer
   f. Presentation layer
   g. Application layer
5. At which OSI layer is the actual configuration of the networking media defined?
   a. Physical layer
   b. Data Link layer
   c. Network layer
   d. Transport layer
   e. Session layer
   f. Presentation layer
   g. Application layer
6. At which OSI layer is network redirection completed?
   a. Physical layer
   b. Data Link layer
   c. Network layer
   d. Transport layer

e. Session layer

f. Presentation layer

g. Application layer

7. At which OSI layer are physical addresses used to send data?

   a. Physical layer

   b. Data Link layer

   c. Network layer

   d. Transport layer

   e. Session layer

   f. Presentation layer

   g. Application layer

8. At which OSI layer is ongoing communication controlled?

   a. Physical layer

   b. Data Link layer

   c. Network layer

   d. Transport layer

   e. Session layer

   f. Presentation layer

   g. Application layer

9. At which OSI layer are TCP/IP addresses used?

   a. Physical layer

   b. Data Link layer

   c. Network layer

   d. Transport layer

   e. Session layer

   f. Presentation layer

   g. Application layer

10. All network operating systems allow you to run applications such as word processors on them. True or False?

11. When one layer communicates with its counterpoint on another system, it does so directly. True or False?

12. Which of the following is *not* a startup type for Windows 2000 Server services?

    a. Automatic

    b. Manual

    c. Dependent

    d. Disabled

13. Once a service has been disabled, which user group(s) can still connect to the service?
    a. Administrators and Account Operators
    b. Administrators and Service Operators
    c. Administrators only
    d. Administrators and Server Operators
14. Changing the background services optimization to application optimization does not require rebooting the server. True or False?
15. You access the Windows 2000 Server service configuration options through the Services Control Panel. True or False?
16. Of the three main Windows 2000 Server-supported protocols, only ______________ and ______________ are routable.
17. Of the three main Windows 2000 Server-supported protocols, ______________ is the fastest and easiest to configure.
18. Windows 2000 Server domains and Windows NT domains are the same. True or False?
19. All operating systems support symmetric multiprocessing. True or False?
20. NetBEUI is one of the routable protocols supported by Windows 2000 Server. True or False?

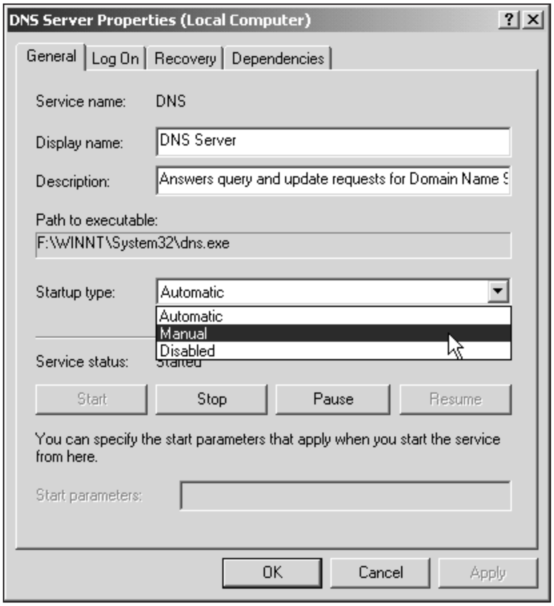
HANDS-ON PROJECTS

Project 8-1

This exercise demonstrates the steps involved in modifying the startup type of a service.

To change the startup type of a service:

1. Click **Start**, **Programs**, **Administrative Tools**, **Component Services**.
2. Highlight the **Services (Local)** option in the left pane (refer to Figure 8-2 earlier in the chapter).
3. Double-click the **DNS Server** service.
4. From the Startup type drop-down menu, choose the **Manual** option (see Figure 8-12).
5. Click the **Apply** button to implement the changes.
6. Click **OK** to close the DNS Server Properties dialog box.
7. Close the **Component Services** window.


DNS Server Properties (Local Computer)

Service name: DNS

Display name: DNS Server

Description: Answers query and update requests for Domain Name S

Path to executable:
F:\WINNT\System32\dns.exe

Startup type: Automatic

Service status: Started

You can specify the start parameters that apply when you start the service from here.

Start parameters:

Figure 8-12 Setting the Startup type for the DNS Server service

Project 8-2

This exercise demonstrates the steps involved in stopping and restarting a service.

To stop and start a service:

1. Click **Start**, **Programs**, **Administrative Tools**, **Component Services**.
2. Highlight the **Services (local)** option in the left pane.
3. Double-click the **DNS Server** service.
4. Click the **Stop** button (see Figure 8-13). The DNS Server service is now stopped.
5. Click the **Start** button to restart the service.
6. Click **OK** to close the DNS Server Properties dialog box.
7. Close the **Component Services** window.

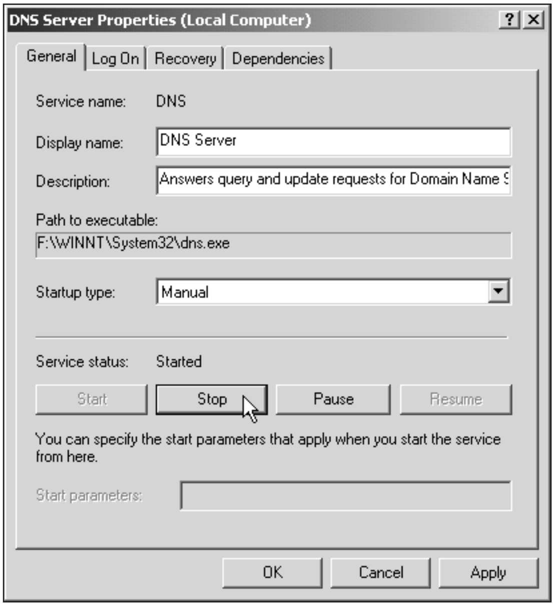
Project 8-3

This exercise demonstrates the steps involved in pausing and restarting a service.

To pause and restart a service:

1. Click **Start**, **Programs**, **Administrative Tools**, **Component Services**.
2. Highlight the **Services (Local)** option in the left pane.
3. Double-click the **DNS Server** service.
4. Click the **Pause** button. The DNS Server service is now paused.

5. Click the **Resume** button. The DNS Server service begins running again.
6. Click **OK** to close the DNS Server Properties dialog box.
7. Close the **Component Services** window.



Service name: DNS

Display name: DNS Server

Description: Answers query and update requests for Domain Name S

Path to executable:
F:\WINNT\System32\dns.exe

Startup type: Manual

Service status: Started

You can specify the start parameters that apply when you start the service from here.

Start parameters:

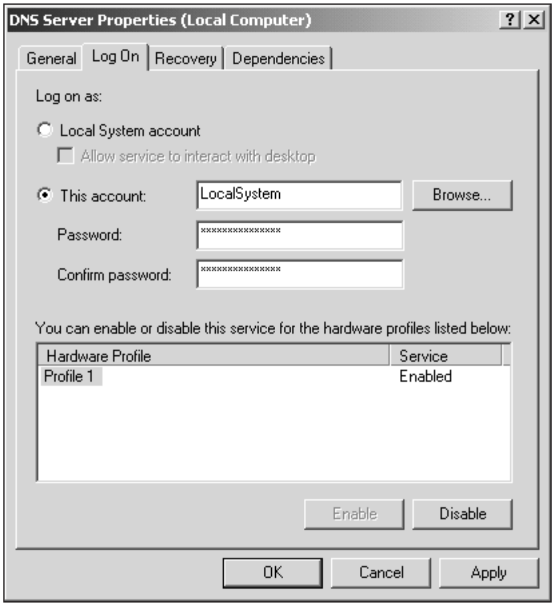
Figure 8-13 Stopping the DNS Server service

Project 8-4

This exercise demonstrates the steps involved in modifying the logon account used by a service.

To change the logon account of a service:

1. Click **Start**, **Programs**, **Administrative Tools**, **Component Services**.
2. Highlight the **Services (Local)** option in the left pane.
3. Double-click the **DNS Server** service.
4. Click the **Logon** tab.
5. Click the **This account** radio button (see Figure 8-14).
6. Click the **Browse** button and select the desired user account, by either highlighting the user's name in the list, or typing it in the Name field (see Figure 8-15).



DNS Server Properties (Local Computer)

Log on as:

○ Local System account

□ Allow service to interact with desktop

● This account: LocalSystem

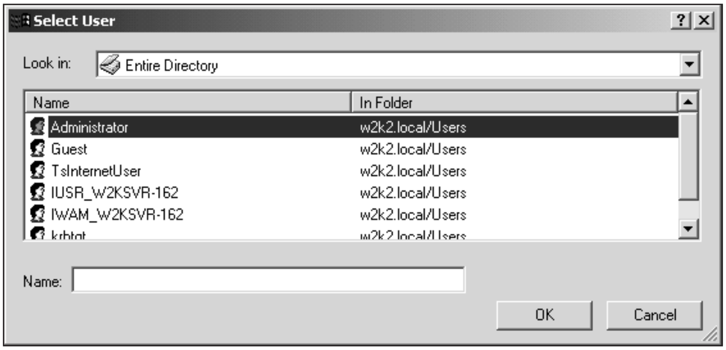
Password: ***************

Confirm password: ***************

You can enable or disable this service for the hardware profiles listed below:

Hardware Profile	Service
Profile 1	Enabled

Figure 8-14 Changing the logon account for the DNS Server service



Select User

Look in: Entire Directory

Name	In Folder
Administrator	w2k2.local/Users
Guest	w2k2.local/Users
TsInternetUser	w2k2.local/Users
IUSR_W2KSVR-162	w2k2.local/Users
IWAM_W2KSVR-162	w2k2.local/Users
krbtgt	w2k2.local/Users

Name:

Figure 8-15 Selecting the user to log on to a service

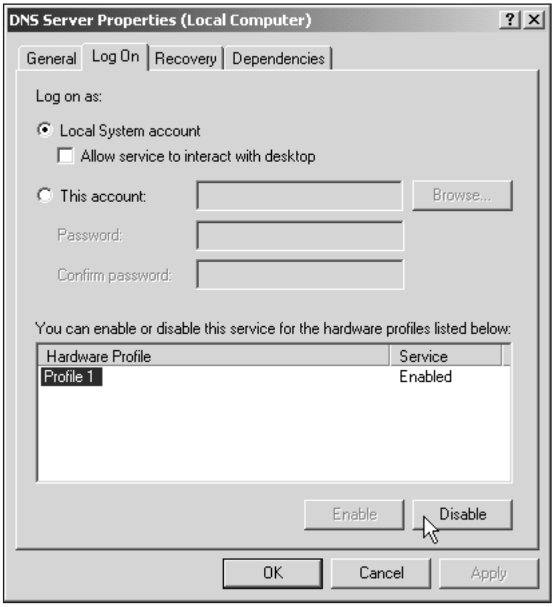
7. Enter the password for the user.
8. Reenter the password for confirmation.
9. Click the **Apply** button to implement the changes.
10. Click **OK** when the confirmation dialog box appears.
11. Click **OK** when the system informs you that the computer must be restarted for the changes to take effect.
12. Click **OK** to close the DNS Server Properties dialog box.
13. Close the **Component Services** window.

Project 8-5

This exercise demonstrates the steps involved in disabling a service in a specific profile.

To disable a service:

1. Click **Start**, **Programs**, **Administrative Tools**, **Component Services**.
2. Highlight the **Services (local)** option in the left pane.
3. Double-click the **DNS Server** service.
4. Click the **Logon** tab.
5. In the lower part of the window, highlight the profile in which you would like this service disabled (see Figure 8-16).



DNS Server Properties (Local Computer)

Log on as:

● Local System account

□ Allow service to interact with desktop

○ This account:

Password:

Confirm password:

You can enable or disable this service for the hardware profiles listed below:

Hardware Profile	Service
Profile 1	Enabled

Figure 8-16 Disabling a service

6. Click the **Disable** button.
7. Click the **Apply** button to implement the changes.

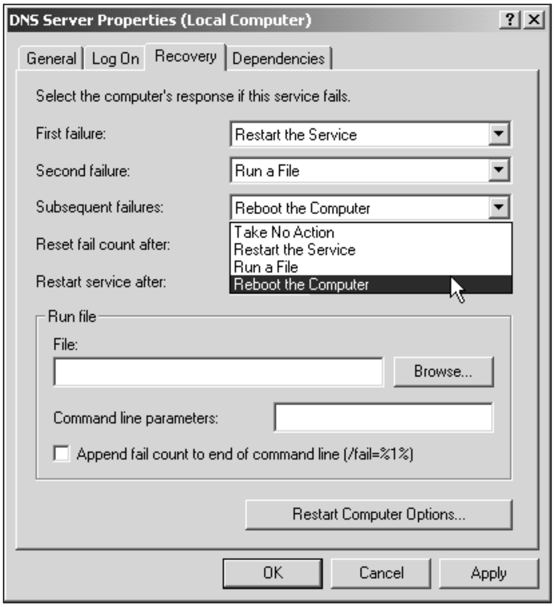
Project 8-6

This exercise demonstrates the steps involved in configuring the recovery options of a service. At the first failure, we attempt to restart the service. After the second failure, we execute a program to page an administrator. (This choice assumes that a modem is connected and configured on the server, and that the paging software is installed in the C:\Pager folder.)

To change the recovery options of a service:

1. Click **Start**, **Programs**, **Administrative Tools**, **Component Services**.

2. Highlight the **Services (local)** option in the left pane.
3. Double-click the **DNS Server** service.
4. Click the **Recovery** tab.
5. From the First failure: drop-down menu, choose the **Restart the service** option.
6. From the Second Failure: drop-down menu, choose the **Run a File** option.
7. Browse to the C:\Pager\Page.exe program.
8. Enter any required information for the Page.exe program (such as the pager phone number).
9. From the **Subsequent failures** drop-down menu, choose the **Reboot the Computer** option (see Figure 8-17).
10. Click on the **Restart Computer Options** button and enter the time before system shutdown and an optional message to send to connected users to notify them of the shutdown.
11. Click on the **Apply** button to set the changes.
12. Click **OK** to close the DNS Server Properties dialog box.
13. Close the **Component Services** window.



Select the computer's response if this service fails.

First failure: Restart the Service

Second failure: Run a File

Subsequent failures: Reboot the Computer

Reset fail count after:

Restart service after:

File:

Command line parameters:

□ Append fail count to end of command line (/fail=%1%)

Figure 8-17 Configuring the recovery options of a service

Case Projects

1. Your organization is looking for a protocol that will allow it to connect its multiple offices. Each of the offices will also be connected to the Internet. Which protocol would you recommend?
2. Your current network uses NetBEUI as its only protocol. Your organization has merged with a second company in another location in the city. Will this protocol work? If not, then which ones will?
3. You have been asked to install a small network for one of your clients. It will run a peer-to-peer network and your client would like to become connected in the least painful and fastest way. Which protocol would you recommend?
4. A client has multiple sites and wants to connect them via Frame Relay connections. Which protocols can be used to accomplish this goal? Which would you recommend and why?